

# Informe Técnico del Sistema de Rutas Óptimas

---

Autor: Carlos Ortega

Fecha: Julio 2025

## Índice

- - Introducción
- - Arquitectura del Sistema
- - Diseño de la Base de Datos
- - Lógica del Grafo y Cálculo de Rutas
- - Generación y Visualización del Grafo
- - Interfaz de Usuario
- - Pruebas y Validación
- - Conclusiones

## 1. Introducción

En este informe se detalla el desarrollo de una aplicación web para el cálculo de rutas óptimas entre ciudades de Ecuador. Se ha utilizado Flask en el backend, MySQL como base de datos, NetworkX para el algoritmo de rutas y Matplotlib para renderizar el grafo. Se incluye además la opción de forzar el paso por ciudades costeras.

## 2. Arquitectura del Sistema

### 2.1. Flujo de Petición-Respuesta

- El usuario envía un POST con origen, destino y opción "Forzar costa".
- Flask recibe la petición en grafo\_routes.py.
- grafo\_controller.py valida datos y construye lista de aristas.
- grafo\_logica.py invoca NetworkX para calcular la ruta.
- Se genera un PNG con Matplotlib y se devuelve al cliente.

### 3. Diseño de la Base de Datos

#### 3.1. Tabla Ciudad

- id (PK, INT)
- nombre (VARCHAR)

#### 3.2. Tabla Ruta

- id (PK, INT)
- origen\_id (FK → Ciudad.id)
- destino\_id (FK → Ciudad.id)
- costo (INT)

### 4. Lógica del Grafo y Cálculo de Rutas

#### 4.1. Construcción del Grafo

```
G = nx.DiGraph()

# Añadir todas las ciudades como nodos

for ciudad in ciudades_bd:

    G.add_node(ciudad.nombre)

# Añadir aristas con peso

G.add_weighted_edges_from(edges)
```

#### 4.2. Algoritmo de Dijkstra

- nx.dijkstra\_path(G, origen, destino)
- nx.dijkstra\_path\_length(G, origen, destino)

#### 4.3. Forzado de Paso por Zona Costera

```
ciudades_costeras = ['Manta', 'Portoviejo', 'Guayaquil']

if costa and not ruta_contiene_costera(ruta_final):

    for costa in ciudades_costeras:

        ruta1 = dijkstra(origen, costa)

        ruta2 = dijkstra(costa, destino)
```

```
if ruta1 and ruta2:  
    ruta_final = ruta1[:-1] + ruta2  
    costo = costo1 + costo2  
    break
```

## 5. Generación y Visualización del Grafo

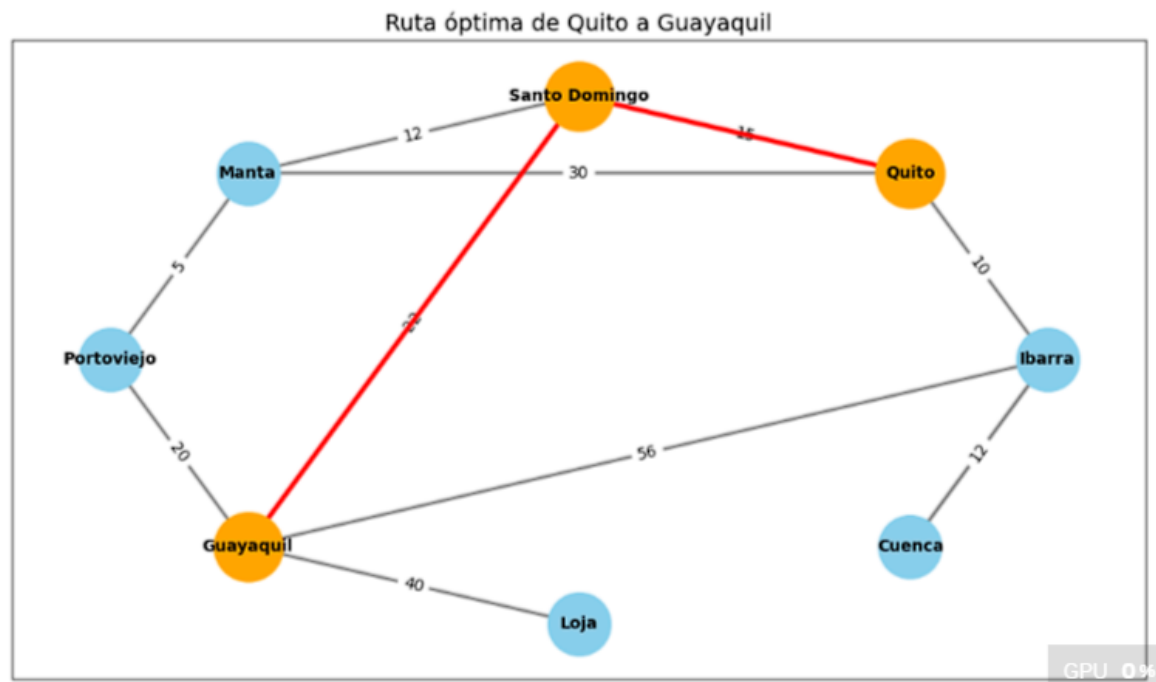
### 5.1. Layout

Se utiliza `kamada_kawai_layout` para una distribución más balanceada:

```
pos = nx.kamada_kawai_layout(G)
```

### 5.2. Renderizado con Matplotlib

```
plt.figure(figsize=(12, 8))  
nx.draw_networkx_nodes(G, pos, node_color='skyblue', node_size=1800)  
nx.draw_networkx_edges(G, pos, edge_color='gray', arrows=True)  
nx.draw_networkx_labels(G, pos, font_size=12)  
# Destacar la ruta óptima en rojo  
nx.draw_networkx_edges(G, pos, edgelist=edge_list, edge_color='red', width=4)  
plt.axis('off')  
plt.tight_layout(pad=2.0)  
plt.savefig(ruta_archivo)
```



## 6. Interfaz de Usuario

- Panel izquierdo superior: CRUD completo de ciudades.
- Panel izquierdo inferior: Tabla de rutas fijas.
- Panel derecho superior: Formulario "Calcular ruta" + checkbox "Forzar costa".
- Panel derecho inferior: Imagen del grafo, ruta tomada y costo total.

Bienvenido

**Ciudades Registradas**

Nueva ciudad  Agregar

ID	Nombre	Acciones
9	Cuenca	
6	Guayaquil	
1	Ibarra	
8	Loja	
4	Manta	

**Rutas Existentes**

Origen	Destino	Costo
Ibarra	Quito	10
Quito	Santo Domingo	15
Quito	Manta	30
Santo Domingo	Manta	12
Manta	Portoviejo	5
Portoviejo	Guayaquil	20

**Calcular Ruta Óptima**

☐ Forzar costa
 Calcular

☒ Ruta generada con éxito

**Ruta tomada:** Quito → Santo Domingo → Guayaquil  
**Costo total:** 37

☒ Visualización de la Ruta

Ruta óptima de Quito a Guayaquil

GPU 0%

## 7. Pruebas y Validación

### 7.1. Escenarios Clave

| Origen | Destino | Normal Ruta | Costo | Costa Ruta | Costo |

|-----|-----|-----|-----|-----|-----|

| Quito | Guayaquil | Quito → Santo Domingo → Guayaquil | 37

| Quito → Santo Domingo → Manta → Portoviejo → Guayaquil | 52 |

| Cuenca | Loja | Cuenca → Ibarra → Quito → Santo Domingo → Guayaquil → Loja | 99

| Cuenca → Ibarra → Quito → Manta → Portoviejo → Guayaquil → Loja | 117 |

## 8. Conclusiones

Se ha implementado con éxito un sistema modular que permite:

- Gestión de ciudades (CRUD)
- Rutas predefinidas y establecidas en el dominio
- Cálculo dinámico de rutas óptimas con NetworkX
- Opción de forzar paso por zonas costeras
- Visualización profesional del grafo en PNG